Closed Queueing Network Models of Interacting Long-Lived TCP Flows

Michele Garetto, *Student Member, IEEE*, Renato Lo Cigno, *Member, IEEE*, Michela Meo, *Member, IEEE*, and Marco Ajmone Marsan, *Fellow, IEEE*

***Abstract*—This paper presents a new analytical model for the estimation of the performance of TCP connections. The model is based on the description of the behavior of TCP in terms of a closed queueing network. The model is very accurate, deriving directly from the finite state machine description of the protocol. The assessment of the accuracy of the analytical model is based on comparisons against detailed simulation experiments developed with the *ns-2* package. The protocol model interacts with an IP network model that can take into account meshed topologies with several bottlenecks.**

Numerical results indicate that the proposed closed queueing network model provides accurate performance estimates in all situations. A novel and interesting property of the model is the possibility of deriving ensemble distributions of relevant parameters, such as, for instance, the transmission window size or the timeout probability, which provide useful insight into the protocol behavior and properties.

***Index Terms*—Analytical models, performance analysis, protocol modeling, queueing networks, TCP.**

I. Introduction and Paper Contribution

TRANSMISSION Control Protocol (TCP) is by far the most widely used technique for the end-to-end control of the information transfer in packet networks. This widespread diffusion is a result of the capability of TCP to adapt to networks with very high bandwidth-delay products and with a huge number of connections per link. Nevertheless, the extreme conditions in which TCP is likely to be used in a few years (terabit per second routers and channels, and millions of connections sharing one physical link) require a careful investigation, at least to support the design and planning of large network segments with high traffic loads.

The pace of research in TCP modeling has been almost frenetic in recent years, as indicated by the large number of high-quality papers that appeared in the literature (see Section I-A for a discussion). Why then another model of TCP? The answer is that some fundamental aspects of the behavior of TCP connections still require investigation. Examples are: 1) the interaction of TCP with multiple bottleneck links; 2) the sensitivity of TCP connections to correlated losses; 3) the transfer delay of short files; 4) the TCP behavior when losses occur and the round-trip time (RTT) estimation is unreliable (see the Karn algorithm [1]); and 5) the effect of the interaction of thousands of connections.

Two are the reasons why available models do not allow the investigation of those aspects: 1) important details of either the protocol operations or the interaction among TCP connections are neglected in the model development, mainly to obtain an acceptable model complexity; and 2) the description of the allocation of the resources in the underlying IP network is either too simple, or embedded in the protocol model.

The main contributions of this paper are the following.

- The protocol description in the model is detailed, but the model solution is very simple.
- The performance estimates produced by the model are extremely accurate: even the window size *distribution* at TCP transmitters is accurately estimated.
- The model solution complexity is *independent* of the number of interacting TCP connections.
- The description of the allocation of resources in the underlying IP network is decoupled from the protocol description, so that complex scenarios, such as multiple bottlenecks, can be modeled with the desired accuracy. The IP network can be modeled with any suitable means; however, we show that when the protocol model is accurate enough and takes into account TCP induced correlations, even a simple IP network model based on uncorrelated queues yields accurate results.

A. Background and Related Work

The literature on TCP modeling is vast, so that it is impossible to provide here a comprehensive overview of previous contributions; rather, we only refer to those papers that are most related to our work.

In [2], the principles for the development of closed queueing network models of window protocols were introduced, explaining in detail the solution technique, the derivation of performance metrics and the limitations of the method; however, no model of a *real* protocol was developed. The approach we adopt in this paper is a direct application of that presented in [2]. In [3] we presented a first application of the technique to TCP connections.

The work in [4] and [5] presents a Markov reward model of several TCP versions on lossy links. The proposed approach consists in modeling the TCP behavior in terms of transmission cycles defined between two loss events, that represent renewal points of the process: Several performance metrics can

Manuscript received December 20, 2000; revised January 30, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. V. Lakshman. This work was supported by the Italian Ministry for Education, University and Research through the MQOS Project.

M. Garetto, M. Meo, and M. Ajmone Marsan are with the Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy (e-mail: garetto@polito.it; michela@polito.it; ajmone@polito.it).

R. Lo Cigno was with the Politecnico di Torino, 10129 Torino, Italy. He is now with the Dipartimento di Informatica e Telecomunicazioni, Università di Trento, 38050 Trento, Italy (e-mail: locigno@dit.unitn.it).

Digital Object Identifier 10.1109/TNET.2004.826297

be derived as rewards during these cycles. The loss process is assumed to be independent from the load offered by the TCP connections and implicitly forbids the study of interacting TCP connections.

Also the approach followed in [6] is based on the exploitation of renewal points, namely of an alternating renewal process. The efficiency of the TCP protocol in exploiting a single bottleneck is estimated with heuristic considerations disregarding the transient behavior of TCP (slow start, timeouts). As in other similar models, the extension to multiple bottlenecks is difficult, and any nonlinear behavior of the protocol cannot be considered, since it would destroy the renewal properties of the model.

The approach adopted in [7] can cope with correlated losses, and it is actually based on the idea that packets are lost in bursts that cover a whole transmission window fraction; the loss rate and RTT are needed as inputs in order to allow the model to compute the TCP throughput and average window size. Results obtained with this model are compared against actual measurements.

A somewhat similar modeling approach can be found in [8], where the performances of TCP-Reno and TCP-Vegas are compared, modeling the two versions through a fluid-based, differential equation approach.

In [9]–[11], the adopted modeling technique is based on the description of the window evolution through differential equations. This modeling approach proves to be powerful enough to describe different TCP versions, some TCP enhancements, and a generalized version of window based congestion control. Also based on differential equations is the model proposed in [12], which focuses on Random Early Detection (RED) routers, assuming that slow start phases and timeouts can be neglected.

In [13] and [14] the authors propose a technique based on differential stochastic equations modeling together the macroscopic behavior of TCP and the network itself. The approach is able to cope with multiple connections and multiple bottlenecks, but does not model the slow start phase and correlated losses, assuming that buffers are actively managed with a RED algorithm.

Another line of research is represented by [15] and [16]. The evolution of TCP is represented here in terms of Max-Plus algebra.

The approach followed in [17] is based on direct Markovian modeling; its main innovation is the presence of nongreedy connections.

II. Model Description and Solution

Fig. 1 presents a simplified high-level description of the proposed model. The model is split in two parts, the TCP sub-model, which describes, by means of closed queueing networks, the behavior of long-lived TCP connections; and the network sub-model, which focuses on the underlying IP network.

Developing the TCP sub-model, we divide TCP connections into G *groups*; each group is an ensemble of connections that share the same path within the network, hence, experiencing the same overall queueing delay and average loss probability. A generic TCP group i can be seen as a traffic relation between an ingress and an egress router.

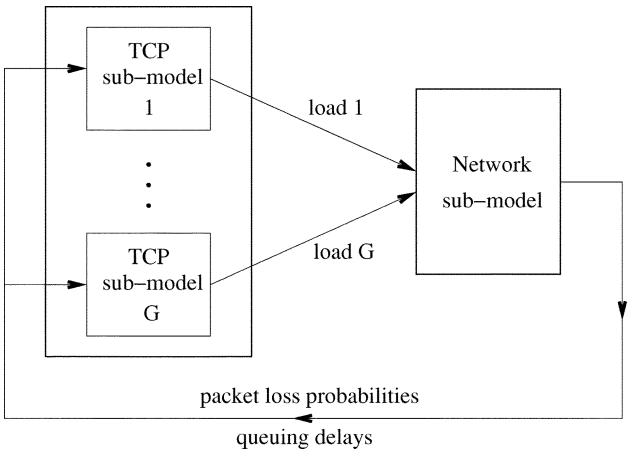

Fig. 1. High-level description of the model.

The TCP sub-model computes the traffic loads Λ_i that *groups* offer to the network. The network sub-model computes the average packet loss probabilities P_{L_i} and average round-trip times $\overline{\mathrm{RTT}}_i$ experienced by *groups*. $\overline{\mathrm{RTT}}_i$ is computed as the harmonic mean of the round-trip times of individual connections [18]. The TCP and the network sub-model interact exchanging the traffic loads offered to the network, and the average packet loss probabilities and average RTTs. The global model solution is obtained with a fixed point procedure, based on the iterative solution of the two sub-models until equilibrium is reached. The fixed point procedure is stopped when the relative error of the estimates of P_{L_i}, $\overline{\mathrm{RTT}}_i$, and Λ_i become smaller than a predefined threshold ϵ.

The fundamental aim of this model is capturing the closed loop interaction of TCP connections with a congested network that reacts dropping packets.

A. TCP Sub-Model

We describe a single TCP group dropping the dependence on the groups (subscript i) and simply using the notation P_L, $\overline{\mathrm{RTT}}$, and Λ.

We model the TCP-Tahoe protocol, as described in [19], assuming that fixed-size TCP segments are transmitted over the underlying IP network. TCP-Tahoe is coherent with the BSD implementation, as well as with the implementation found in the *ns-2* simulator [20] that we use for validation purposes. The description of the TCP protocol is beyond the scope of this paper; the interested reader is referred to [19] or the appropriate IETF RFCs.

The behavior of N concurrent long-lived TCP connections is modeled using a closed network of $M/G/\infty$ queues. Each queue describes a *state* of the TCP protocol; the number of customers in a generic queue Q, N_Q, is the number of TCP connections that are in that specific state.

The modeling approach is extremely flexible, and allows the adoption of an almost arbitrary level of abstraction. We chose to model the protocol in full detail, neglecting only marginal aspects of the protocol operations. The queueing network in Fig. 2 shows the TCP dynamics when the maximum window size is

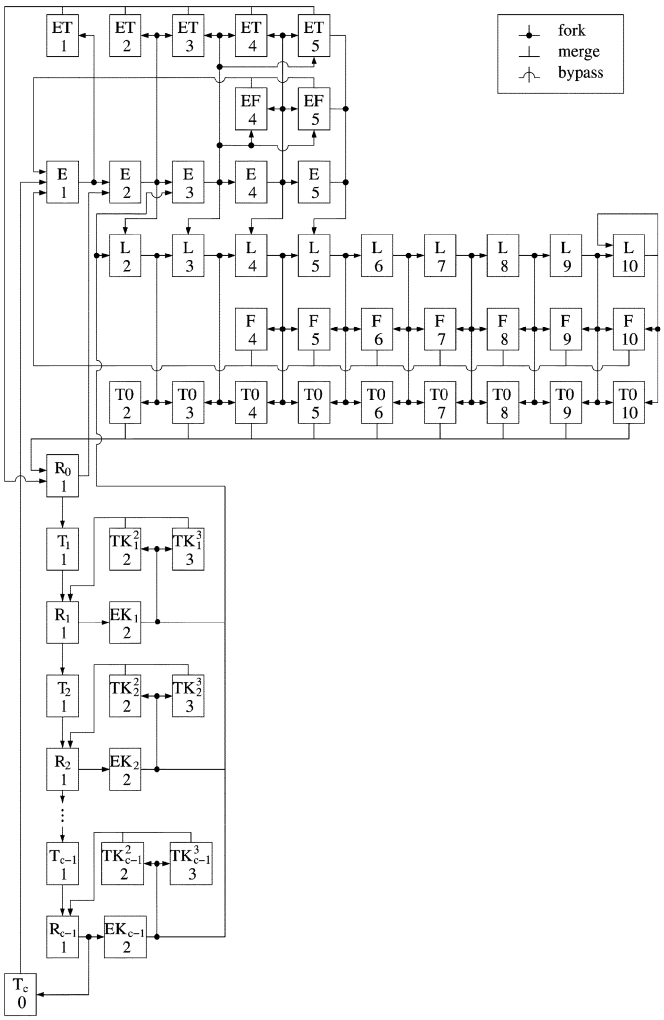

Fig. 2. Queueing network model of TCP-Tahoe.

$W = 10$ segments. Each queue is characterized by the congestion window (*cwnd*) size expressed in number of segments. In addition, the queue describes whether the transmitter is in slow start, congestion avoidance, waiting for a fast retransmit or a timeout. The queues in Fig. 2 are arranged in a matrix pattern: all queues in the same row correspond to similar protocol states, and all queues in the same column correspond to equal window size. Queues below R_0 model backoff timeouts, retransmissions, and the Karn algorithm.

For the specification of the queueing network model, it is necessary to define for each queue Q:

1) the average service time τ_Q, which models the time spent by TCP connections in the state described by Q;
2) the transition probabilities $P(Q_i, Q_j)$, which are the probabilities that TCP connections enter the state described by Q_j after leaving the state described by Q_i.

The queueing network comprises 11 different types of queues, described in some detail below. In Table I, queue types are listed together with their average service time, which, as already mentioned, represents the time a connection spends in a given protocol state. Most of them derive directly from the protocol properties; details about the least intuitive ones can be found in [3].

Queues E_i $(1 \leq i \leq W/2)$ model the exponential window growth during slow start; the index i indicates the congestion window size. As shown in Table I, the average service time depends on the round-trip time through an apportioning coefficient σ that takes into account the fact that during slow start the TCP congestion window grows geometrically with base 2 (see [3] for details).

Queues ET_i $(1 \leq i \leq W/2)$ model the TCP transmitter state after a loss occurred during slow start: the congestion window has not yet been reduced, but the transmitter is blocked because its window is full; the combination of window size and loss pattern forbids a fast retransmit (i.e., less than three duplicate ACKs are received) and the TCP source is waiting for a timeout to expire. The service time is function of the term T_0 which equals $\max(3\ \mathrm{tic}, 4\overline{\mathrm{RTT}})$.[1]

Queues EF_i $(4 \leq i \leq W/2)$ model a situation similar to that of queues ET_i, but the TCP source is waiting for the duplicate ACKs that trigger the fast retransmit.

Queues L_i $(2 \leq i \leq W)$ model the linear growth during congestion avoidance (notice that queue L_1 does not exist).

Queues F_i $(4 \leq i \leq W)$ model losses during congestion avoidance that trigger a fast retransmit.

Queues $T0_i$ $(2 \leq i \leq W)$ model the detection of losses by timeout during congestion avoidance.

Queues T_i $(1 \leq i \leq C)$ model the time lapse before the expiration of the $(i+1)$th timeout for the same segment (backoff timeouts). C is the maximum number of consecutive retransmissions allowed before closing the connection. Queue T_C actually models TCP connections that were closed for an excessive number of timeouts. Closed connections are supposed to re-open after a random time; we chose $\tau_{T_C} = 180\ \mathrm{s}$, however this value has a marginal impact on results.

Queues R_i $(0 \leq i \leq C-1)$ model the retransmission of a packet when the timeout expires.

Queues EK_i $(1 \leq i \leq C-1)$ model the first stage of the slow start phase (i.e., the transmission of the first two non-retransmitted packets) after a backoff timeout.

Queues TK_i^2 $(1 \leq i \leq C-1)$ model the wait for timeout expiration when losses occurred in queues EK_i leaving the congestion window at 2.

Queues TK_i^3 $(1 \leq i \leq C-1)$ model the wait for timeout expiration when losses occurred in queues EK_i leaving the congestion window at three.

The probabilities $P(Q_i, Q_j)$ that customers completing their service at queue Q_i move to queue Q_j can be computed from the dynamics of TCP. These dynamics depend on the working conditions of the network, i.e., they depend on the round-trip time as well as on the packet loss rate. In order to cope with the correlation among losses of packets within the same congestion window,[2] we introduce two different loss probabilities: the prob-

[1]The "tic" is the time granularity of the TCP protocol, i.e., the minimum amount of time that separates non-self-clocked protocol operations.

[2]For the characteristics of TCP, the correlation in packet losses is limited to one congestion window, independently from the window size.

TABLE I
QUEUES AND SERVICE TIMES

Queue		Service time
E_1		$\overline{\text{RTT}}$
E_2		$\overline{\text{RTT}}$
E_i	$i = 2^n \quad n \geq 2,$	$\sigma\,\overline{\text{RTT}}$
E_i	$2^{n-1}+1 \leq i \leq 2^n - 1$	$\frac{(1-\sigma)\overline{\text{RTT}}}{2^{n-1}-1}$
ET_i	$1 \leq i \leq 3$	$T_0 - \overline{\text{RTT}}$
ET_i	$i \geq 4$	T_0
EF_i		$\frac{\overline{\text{RTT}}}{2} + 4\frac{\overline{\text{RTT}}}{i}$
L_i		$\overline{\text{RTT}}$
F_i		$\frac{\overline{\text{RTT}}}{2} + 4\frac{\overline{\text{RTT}}}{i}$
$T0_2$		$T_0 - \overline{\text{RTT}}$
$T0_i$	$3 \leq i \leq W$	$T_0 - \overline{\text{RTT}}/2$
T_i	$1 \leq i \leq 6$	$2^i T_0$
T_i	$7 \leq i \leq C-1$	$64\,T_0$
R_i		$\overline{\text{RTT}}$
EK_i		$\overline{\text{RTT}}$
TK_i^2		$T_i - \overline{\text{RTT}}$
TK_i^3		$T_i - \overline{\text{RTT}}$

ability of loss of the first segment of the active sliding window, P_{L_f} (where f stands for "first"), and the loss probability for any other segment of the same window P_{L_a} (where a stands for "after"). P_{L_f} is also the probability of losing a burst of packets. The derivation of P_L, P_{L_f}, and P_{L_a} is reported in Section II-B; assume, for the moment, that they are known.

Table II reports the transition probabilities $P(Q_i, Q_j)$ from source queue Q_i, to destination queue Q_j, under the conditions indicated in the last column; $\wedge$ is the logical AND operator. In Table II, the following notation is used: $P_S = 1 - P_L$, $P_{S_f} = 1 - P_{L_f}$, and $P_{S_a} = 1 - P_{L_a}$ for the successful packet delivery probabilities; $P_T(i)$ for the probability that the window growth threshold *ssthresh* has value i; $P_C(i) = P\{ssthresh < i\}$ is the cumulative distribution of $P_T(i)$; $Z_{i,j}^T = (P_T(i)/1 - P_C(j))$ and $Z_{i,j}^C = (1 - P_C(i)/1 - P_C(j))$ are quantiles of the above distribution.

Transition probabilities between queues modeling the exponential and the linear congestion window growth follow from the correct delivery of all the packets offered to the network (see $P(E_i, L_i)$ and $P(E_i, E_{i+1})$ in Table II).

Transitions to and from queues associated with timeouts and retransmissions are also relatively easy to compute, though their definition requires a careful account of Karn's algorithm; refer to transitions $P(R_i, \cdot)$ in the table.

Successful and failed transmissions of the first packets after the retransmission, i.e., while the RTT estimates and the timeout value are not updated, cause transitions out of queues EK_i.

TABLE II
TRANSITION PROBABILITIES

Q_i	Q_j	$P(Q_i, Q_j)$	Condition
E_1	E_2	P_{S_f}	
	ET_1	P_{L_f}	
E_2	ET_2	$P_{L_f} + P_{S_f} P_{L_f} Z_{2,2}^T$	
	ET_3	$P_{S_f} P_{L_f} Z_{3,2}^C$	
E_i	E_{i+1}	$P_{S_f}^2 Z_{i+1,i}^C$	$2 \leq i < W/2$
	L_i	$P_{S_f}^2 Z_{i,i}^T$	$2 \leq i \leq W/2$
	ET_j	$P_{L_f} Z_{j,i}^T P_{to}(j-1)$	$i \leq j < 2(i-1)\ \wedge$
	EF_j	$P_{L_f} Z_{j,i}^T P_{ft}(j-1)$	$3 \leq i \leq W/4+1$
	ET_j	$P_{L_f} Z_{j,i}^C P_{to}(j-1)$	$j = 2(i-1)\ \wedge$
	EF_j	$P_{L_f} Z_{j,i}^C P_{ft}(j-1)$	$3 \leq i \leq W/4+1$
	ET_j	$P_{L_f} Z_{j,i}^T P_{to}(j-1)$	$i \leq j \leq W/2\ \wedge$
	EF_j	$P_{L_f} Z_{j,i}^T P_{ft}(j-1)$	$i > W/4+1$
	ET_j	$P_{S_f} P_{L_f} Z_{j,i}^T P_{to}(j-1)$	$i \leq j < 2i-1\ \wedge$
	EF_j	$P_{S_f} P_{L_f} Z_{j,i}^T P_{ft}(j-1)$	$3 \leq i \leq \frac{W/2+1}{2}$
	ET_j	$P_{S_f} P_{L_f} Z_{j,i}^C P_{to}(j-1)$	$j = 2i-1\ \wedge$
	EF_j	$P_{S_f} P_{L_f} Z_{j,i}^C P_{ft}(j-1)$	$3 \leq i \leq \frac{W/2+1}{2}$
	ET_j	$P_{S_f} P_{L_f} Z_{j,i}^T P_{to}(j-1)$	$i \leq j \leq W/2\ \wedge$
	EF_j	$P_{S_f} P_{L_f} Z_{j,i}^T P_{ft}(j-1)$	$i > \frac{W/2+1}{2}$
L_2	$T0_2$	P_{L_f}	
	$T0_3$	$P_{S_f} P_{L_f} + P_{S_f}^2 P_{L_f}$	
L_3	$T0_3$	P_{L_f}	
L_i	$T0_i$	$P_{L_f} P_{to}(i-1)$	$4 \leq i \leq W-1$
	F_i	$P_{L_f} P_{ft}(i-1)$	
	$T0_{i+1}$	$P_{S_f}(1 - P_{S_f}^i) P_{to}(i)$	$3 \leq i \leq W-1$
	F_{i+1}	$P_{S_f}(1 - P_{S_f}^i) P_{ft}(i)$	
	L_{i+1}	$P_{S_f}^{i+1}$	$2 \leq i \leq W-1$
L_W	L_W	$P_{S_f}^W$	
	$T0_W$	$(1 - P_{S_f}^W) P_{to}(W-1)$	
	F_W	$(1 - P_{S_f}^W) P_{ft}(W-1)$	
EK_i	L_2	$P_{S_f}^2 Z_{2,2}^T$	$1 \leq i \leq C-1$
	E_3	$P_{S_f}^2 Z_{3,2}^C$	
	TK_i^2	$P_{L_f} + P_{S_f} P_{L_f} Z_{2,2}^T$	
	TK_i^3	$P_{S_f} P_{L_f} Z_{3,2}^C$	
R_0	E_2	P_{S_f}	
	T_1	P_{L_f}	
R_i	EK_i	P_{S_f}	$1 \leq i \leq C-1$
	T_{i+1}	P_{L_f}	

For transitions that are consequence of losses, we introduce the probability that a loss is detected by timeout given that i segments were transmitted after the lost one:

$$P_{to}(i) = \sum_{j=0}^{2} \binom{i}{j} P_{L_a}^{i-j} P_{S_a}^{j}, \qquad 3 \leq i \leq W$$

and the corresponding probability that the loss is detected by fast retransmit $P_{ft}(i) = 1 - P_{to}(i)$. P_{to} and P_{ft} appear in the transition probabilities $P(L_i, \cdot)$, which correspond to losses

during the congestion avoidance phase. The possibility of losing more than one packet within the congestion window complicates the description, since the destination queue depends on the loss pattern.

Transitions due to losses during slow start are instead quite cumbersome to consider, because the loss pattern and the *ssthresh* distribution (see Section II-A2) heavily influence how much the congestion window is opened before the transmitter detects the loss, either by fast retransmit or by timeout expiration. We distinguish between transitions happening when the first packet transmitted with window size i is lost, and transitions happening when the second packet transmitted with window size i is lost; in the first case the term P_{L_f} appears in the probabilities $P(E_i, ET_j)$ and $P(E_i, EF_j)$, while in the second case the expression of the probability contains $P_{S_f}P_{L_f}$.

1) Load Offered to the Network: The total load Λ offered to the network by all TCP connections is the sum of all the loads Λ_Q offered by connections in the state described by Q. Each connection in queue Q offers to the IP network a number of packets equal to $\mathcal{P}_Q$. The actual load offered to the IP network by each queue is $\Lambda_Q = E[N_Q]\mathcal{P}_Q/\tau_Q = \lambda_Q\mathcal{P}_Q$, where λ_Q is the arrival rate at queue Q. Terms λ_Qs are computed from the queueing network solution assuming a total of N connections.

In queues T_i and ET_1 no packet is generated. One packet is generated per service in queues R_i which model the retransmission of a packet when timeout expires, and in queue E_1, where window size equals 1.

The generation of two packets in queues E_i, with $i > 1$, is due to the exponential window size increase in slow start. Two packets are generated per service also in queues EK_i, standing for the transmission of the first two non-retransmitted packets after a backoff timeout. Similarly, two packets are generated in queues TK_i^3.

More complex is the derivation of the number of packets generated in queues EF_i and ET_i, for $i > 2$, since it depends on which packet was lost in the previously visited queue. It results:

$$\mathcal{P}_{EF_i} = \mathcal{P}_{ET_i} = \left[(i-2)\frac{P_{L_f}}{1-P_{S_f}^2} + (i-1)\frac{P_{S_f}P_{L_f}}{1-P_{S_f}^2}\right].$$

Instead, the load offered by connections in queues L_i is

$$\mathcal{P}_{L_i} = \left[P_{L_f}i + (1-P_{L_f})(i+1)\right], \qquad i \le W-1$$
$$\mathcal{P}_{L_W} = W.$$

The computation of the load offered by individual queues of type F_i and $T0_i$ is cumbersome, since for each queue it depends on the loss pattern. On the contrary, if we consider the aggregate load collectively offered by the set of queues $\mathcal{S}_{Q_{ll}} = \bigcup_i [F_i \cup T0_i]$ then its computation is much easier:

$$\Lambda_{\mathcal{S}_{Q_{ll}}} = \sum_{i=2}^{W}\left[\sum_{j=1}^{i} jP_{S_f}^j P_{L_f}\right]\lambda_{L_i}.$$

Finally, for queues TK_i^2 and ET_2 we can write

$$\mathcal{P}_{TK_i^2} = \mathcal{P}_{ET_2} = \frac{P_{S_f}P_{L_f}Z_{2,2}^T}{P_{L_f} + P_{S_f}P_{L_f}Z_{2,2}^T}.$$

2) Distribution of ssthresh: The threshold (*ssthresh*) that discriminates between the slow start and congestion avoidance phases introduces a memory in the TCP behavior. Indeed, the protocol evolution does not depend only on the present window size, but also on the value of the window at the previous loss event. This implies that a complete description of the protocol would require a number of queues that grows proportionally to W^2, while the model described so far has a number of queues proportional to W. Rather than doing this, we resort to a stochastic approach based on the consideration that the steady-state distribution of the window size, which can be computed from the steady-state distribution of the number of customers in queues, allows the estimation of the distribution of *ssthresh*.

Let us introduce the set of queues visited only after a loss event: $\mathcal{S}_{Q_a} = \bigcup_i [EF_i \cup ET_i \cup T0_i \cup F_i]$ and let $\lambda_a(i)$ be the aggregate arrival rate at queues in $\mathcal{S}_{Q_a}$ that have the same window size i. Then the probability $P_T(i) = P\{ssthresh = i\}$ is

$$P_T(2) = \frac{\sum_{k=1}^{5}\lambda_a(k)}{\sum_{k=1}^{W}\lambda_a(k)}$$
$$P_T(i) = \frac{\lambda_a(2i) + \lambda_a(2i+1)}{\sum_{k=1}^{W}\lambda_a(k)}, \qquad 3 \le i \le \frac{W}{2}.$$

B. Network Sub-Model

The TCP model requires as input the loss probabilities P_{L_f} and P_{L_a}, and the average round-trip time $\overline{\mathrm{RTT}}$ of TCP connections. If these parameters are available, e.g., from network measurements, the TCP model predicts parameters such as the offered load, throughput, time spent in timeout, and window size or *ssthresh* distributions. However, our objective is to predict the performance of concurrent TCP connections given a physical description of the network (i.e., knowing only network span and topology, link capacity, buffer size, etc.), obtaining as results also the loss probability, the average buffer occupancy and the value of $\overline{\mathrm{RTT}}$ as estimated by TCP transmitters. We thus need a model of the interaction of TCP connections with the underlying IP network.

One of the goals of this work is to show that a detailed model of the dynamics of TCP connections coupled with a simple model of the network behavior is capable of producing accurate performance estimates, thus indicating that performance is driven more by the TCP dynamics than by the router behavior. Moreover, this approach allows the solution of complex, multi-bottleneck topologies loaded by several different groups of TCP connections.

1) Router Interface Model: Consider a single router interface with its transmission buffer of B packets and its link of capacity C_P packets/s. In presence of greedy connections that tend to overload the network, the average loss rate is dominated by the excess load offered by connections with respect to the bottleneck capacity, so that different queueing models provide similar estimates of the average loss probability. Therefore, we

use a simple $M/M/1/B$ queue for each link on the topology. Other approaches with significantly greater complexity (mainly based on group arrivals and services) were tested for the sake of completeness, but results do not change significantly in the case of long-lived flows, substantiating the idea that the average loss rate of greedy connections is dominated by the protocol behavior, that defines the excess load.

Losses in IP networks do not follow Bernoulli patterns [7], [23] due to the presence of FIFO drop-tail buffers and synchronization among TCP sources.

In [7], the authors assume that the intra-connection correlation among losses is 1, i.e., after a loss, all the remaining packets in a transmission window are also lost, though the first lost segment is not necessarily the first segment in the window. In our case, considering that the TCP behavior is such that the first lost packet in a burst of losses is the one in the lowest position in a window (ACKs make the window slide until this is true) it seems more realistic to assume that the loss rate within a burst is obtained by scaling the initial loss rate, $P_{L_\alpha} = \alpha P_{L_f}$, with the constraint that the average loss rate is respected. Empirical observations show that the loss correlation becomes stronger as the window size grows; to catch this behavior we set $\alpha = \overline{w}$, subject to the above constraint, where $\overline{w}$ is the average congestion window size computed by the TCP model.

Finally, note that we consider here drop-tail buffers only. The analysis of networks with Active Queue Management (AQM) algorithms, such as RED [23], does not pose additional problems. Indeed, when RED is used, the estimation of the average buffer occupancy allows also the estimation of the packet drop probability, and the average buffer occupancy is easy to compute. Moreover, due to the small correlation between losses imposed by AQM schemes, the estimation of the network behavior is simplified. The results for RED buffers are as good as those shown here, or better, and are not reported for the sake of brevity.

2) Multibottleneck Representation: We consider a generic topology loaded by G groups of TCP connections, as shown in Fig. 1. The connection path π_i is the collection of all links traversed by TCP connections in group i. Each buffer is a potential bottleneck whose level of congestion depends on the total number of TCP connections competing for resources on the link.

At the network level, we assume that losses in different buffers are not correlated, which is fairly reasonable, since there is no coordination between routers. Obviously, the connection behaviors are correlated, due to the fact that connections throttled on one link will offer to downstream nodes only the load corresponding to their share on the upstream buffer.

For each TCP group i, the average probability that a packet is successfully transferred through the network is

$$P_{Si} = \prod_{l \in \pi_i} (1 - P_{L_l}) \tag{1}$$

where P_{L_l} is the average drop rate at the buffer of link l.

The load offered to the network by each TCP group is computed as explained in Section II-A1, hence, the load of each link is given by the sum of the loads generated by the groups sharing the link.

C. Solution Complexity and Convergence Properties

The model solution is obtained by iterating between the network sub-model and the TCP sub-model with a fixed point algorithm, until convergence is reached.

Since the network sub-model solution simply consists in computing the closed-form formulas of the M/M/1/B queues, the complexity of each step of the iterative procedure is dominated by the solution of the TCP sub-model. Since the closed queueing network describing the TCP behavior is composed of infinite-server queues, the queueing network solution is not dependent on the number of customers and reduces to the computation of the average arrival rates λ_Qs. In order to derive the λ_Qs, a system of linear equations has to be solved; employing standard techniques, the complexity of this step is $O(M_Q^3)$, where M_Q is the total number of queues. Moreover, exploiting the almost triangular structure of the system of linear equations, the complexity reduces to $O(M_Q^2)$. Since M_Q is of the order of few hundreds, the CPU time required for each step of the fixed point algorithm is extremely small (less than a second on any modern PC running Linux).

The number of iterations before convergence depends on the accuracy ϵ. Setting the accuracy $\epsilon = 10^{-6}$, the number of iterations ranges from 20–30 to 200–300 for the most critical cases (for very small or very large number of connections). In conclusion the time required to solve the model is almost independent from the number of modeled connections.

The comparison with *ns-2* simulations is striking. The CPU time of any simulation increases linearly with the number of simulated events, thus we can expect that, in order to reach a given accuracy per connection, the CPU time increases linearly with the number of connections. However, also the memory requirements of the simulation program increases roughly linearly with the number of simulated connections, with the effect that the simulation process makes more and more paging operations per simulated event. We empirically observed CPU times increase roughly quadratically with the number of connections, up to a point where memory requirements exceed the machine capacity. For the results presented in this paper, the CPU time *for each simulation point* ranges from 5–6 minutes up to several hours. The speedup obtained with the model is at least two orders of magnitude in the case of very few connections up to nearly four orders of magnitude in the case of hundreds of connections. The gain in the case of thousands of connections cannot be quantified since simulations cannot be run.

III. Model Validation and Numerical Results

In order to validate our analytical model of TCP, we compare the performance predictions obtained from the queueing network solution against point estimates and 95% confidence intervals obtained from very detailed simulation experiments (in the cases where these are possible). The tool used for simulation experiments is *ns version* 2 [20]; confidence intervals were obtained with the "batch means" technique, using 30 batches.

A. "User-Centric" Topology of the Internet

Most of the analytical studies of TCP consider a single bottleneck topology as a significant setup for the assessment of the

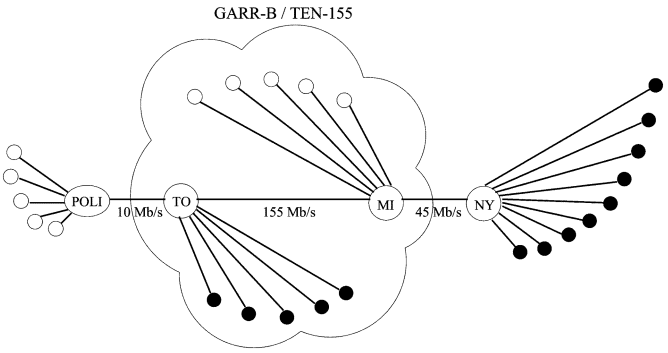

Fig. 3. Abstract representation of the Internet as seen from hosts connected to the LAN of Politecnico di Torino; white circles represent TCP clients; black circles represent TCP servers.

accuracy of approximate models. The reason is often the impossibility of considering more complex topologies, as well as the objective difficulty of finding "general" meshed topologies and not only sample topologies. We chose a somewhat different approach for the validation of the proposed queueing network model of TCP.

To start with, we consider a networking environment which closely resembles the actual path followed by Internet connections from our University LAN to Web sites in Europe and the USA. This is a "user-centric" approach, since we do not try to represent the Internet "as it is," but as the users in our institution perceive it: the presence of a bottleneck in a portion of the Internet which is rarely or never visited has no influence on the perceived performance.

In Section III-G, instead, we show results for an arbitrary meshed topology; however, this topology does not represent any particular portion of the Internet, or any particular "view" of it.

The topology of the network that we consider in this first analysis is shown in Fig. 3; at the far left we can see a set of terminals connected to the internal LAN of Politecnico di Torino. These terminals are the clients of the TCP connections we are interested in (white circles in the figure represent TCP clients; black circles represent TCP servers). The distance of these clients from the Politecnico router is assumed to be uniformly distributed between 1 and 10 km. The LAN of Politecnico is connected to the Italian IP network for universities and research institutions, named GARR-B (a portion of the European TEN-155 academic network), through a 10-Mb/s link whose length is roughly 50 km (this link will be called POLI-TO). Internally, the GARR-B/TEN-155 network comprises a number of routers and 155-Mb/s links. One of those connects the router in Torino with the router in Milano; its length is set to 100 km. Through the GARR-B/TEN-155 network, clients at Politecnico can access a number of servers, whose distance from Politecnico is uniformly distributed between 100 and 6800 km. From Milano, a 45-Mb/s undersea channel whose length is about 5000 km reaches New York, and connects the GARR-B network to the North American Internet backbone (this link will be called MI-NY). Many other clients use the router in Milano to reach servers in the U.S. The distance of those clients from Milano is assumed to be uniformly distributed between 200 and 2800 km. Finally, the distance of

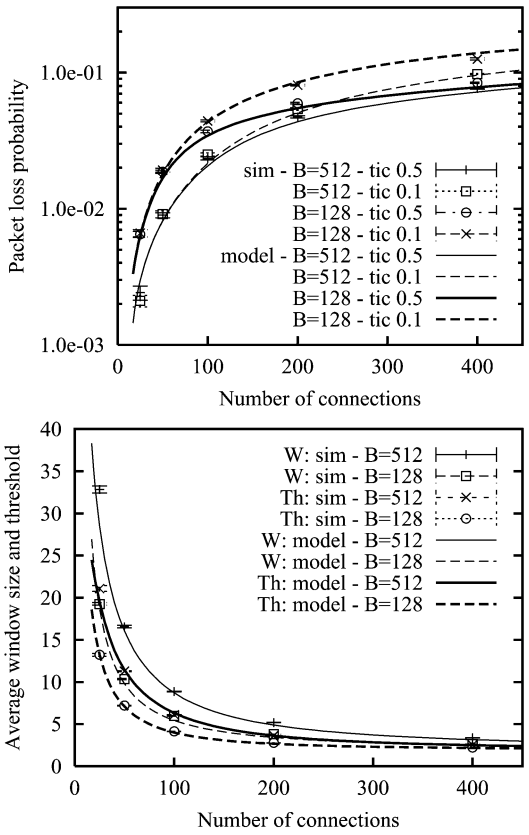

Fig. 4. Packet loss probability versus number of connections (upper plot); average window and *ssthresh* for 100-ms tic (lower plot).

servers in the U.S. from the router in New York is assumed to be uniformly distributed between 200 and 3800 km.

For simplicity, in our performance study we shall restrict our interest to three types of TCP connections:

1) TCP connections for the transfer of Web pages from U.S. Web sites (servers) to users at Politecnico di Torino (clients);
2) TCP connections for the transfer of Web pages from Italian and European servers to clients at Politecnico di Torino;
3) TCP connections for the transfer of Web pages from U.S. Web sites to users connected to the GARR-B/TEN-155 network through the MI-NY link.

We shall assume that congestion in the network can be due to the overload of either the 10-Mb/s POLI-TO channel (which is crossed by connections of types 1 and 2) or the 45-Mb/s MI-NY channel (which is crossed by connections of types 1 and 3). We do not consider the possibility of congestion of the 155-Mb/s GARR-B channels. We assume that 25% of the connections traversing the POLI-TO channel also use the MI-NY link.

The packet size is constant, equal to 1024 bytes; the maximum window size is assumed to be 64 packets. We consider the cases of buffer sizes equal to either 128 or 512 packets, and TCP tic values equal to either 100 or 500 ms.

From the estimation of the loss probabilities over the POLI-TO and MI-NY channels, we can estimate the total loss probability of TCP connections for the transfer of Web pages from U.S. Web sites to users at Politecnico di Torino according to (1) as $P_L =$

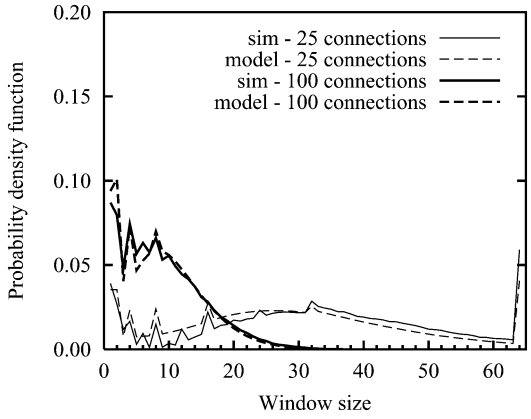

Fig. 5. Window distribution when the bottleneck is the MI-NY link with $B = 512$ in case of 25 and 100 competing connections.

$1-P_S = P_{L_1}+P_{L_2}-P_{L_1}P_{L_2}$, where P_{L_1} is the loss probability on link POLI-TO and P_{L_2} is the loss probability on link MI-NY. The overall model is solved as explained in Sections II-B and II-C.

Instead, the average window size value for the same TCP connections can be calculated from the estimation of the total loss probability and total RTT, by using the queueing network model of TCP with fixed values of loss probability and RTT.

B. Congestion on the MI-NY Channel Only

Let us first consider the case when only the MI-NY channel is overloaded. Fig. 4 contains two separate plots, that show (top plot) the analytical and simulation estimates for the packet loss probability, as a function of the number of connections traversing the link, for the different considered buffer sizes (either 128 or 512 packets) and TCP tic values (either 100 or 500 ms), and (bottom plot) the average TCP window value and the corresponding average threshold value for TCP tic equal to 100 ms, as a function of the number of connections traversing the link, for the different considered buffer sizes (either 128 or 512 packets). The analytical estimates can be observed to predict with very good accuracy the results produced by simulation experiments. Window size and threshold for tic 500 ms follow the same behavior (and accuracy) as those with tic 100 ms, and are not reported for the sake of brevity.

A similar accuracy of the analytical estimates can be observed also for other performance parameters. For example, in Fig. 5 we show the curves of the distribution of the TCP window size obtained from the model and the simulator in the case of 512 packets buffers and TCP tic equal to 500 ms, for two different numbers of connections crossing the MI-NY channel: either 25 or 100. The fact that the queueing network model is capable of providing accurate estimates not only for average values, but also for distributions is a remarkable indication of the fact that it actually captures most of the internal dynamics of the TCP protocol. Indeed, the spikes shown by the window size distribution in Fig. 5 for values 4, 8, 16, and 32 correspond to the congestion window dimensions that are most likely during the slow start phase, when the window is doubled every RTT. Since TCP tends to cluster packet transmission at the beginning of a cycle when RTT is larger than the window transmission time,

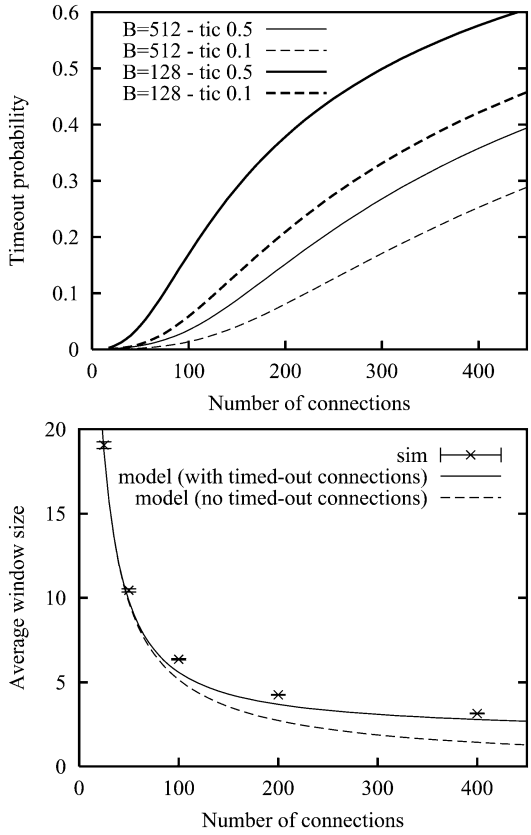

Fig. 6. Fraction of time spent waiting for a timeout to expire versus the number of connections (upper plot) and comparison of the average window size computed considering or neglecting the window size while waiting for timeout expiration with buffer $B = 128$ (lower plot).

the result is that the protocol spends much more time in states corresponding to window sizes that are powers of two, while, during slow start, it quickly skips the states with window size that is not a power of two.

The model also allows the evaluation of performance indices whose computation may be difficult within a simulation. As an example, we show in Fig. 6 the curves of the timeout probability, which is the fraction of TCP connections that are waiting for a timeout to expire, versus the number of connections traversing the link, for the different considered buffer sizes (either 128 or 512 packets) and TCP tic values (either 100 or 500 ms) (upper plot), as well as a comparison between the average window values that result by either considering or neglecting the window value of TCP connections experiencing a timeout, versus the number of connections traversing the link (lower plot). It must be noted that TCP implementations, as well as the code available in *ns-2*, provide a window value for TCP connections experiencing a timeout which is the one that the connection had before the timeout; however, in spite of this window value, the timed-out connection cannot transmit any packet during the timeout. This means that the average traffic offered by a TCP connection to the network cannot be obtained as the ratio of the average window size over the RTT. On the contrary, if we consider the value of the average window size only for non-timed-out TCP connections (by setting to zero the window size of timed-out connections) it is possible to infer from it the average traffic offered by a TCP connection

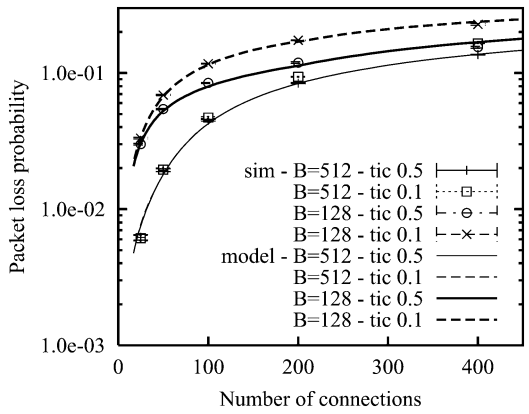

Fig. 7. Packet loss probability versus number of connections when the bottleneck is the POLI-TO link.

to the network as the ratio of such average window size over the RTT. These results show interesting behaviors, indicating that the probability of experiencing a timeout is quite sensitive to the number of connections and their parameters, and that, when the number of connections traversing the link is high, the average window size values obtained excluding timeouts are approximately half of those obtained including connections experiencing a timeout.

C. Congestion on the POLI-TO Channel Only

By considering congestion only on the POLI-TO channel, it is possible to derive results equivalent to those shown for the MI-NY channel. For the sake of brevity we only report here in Fig. 7 the results equivalent to those previously shown in the top plot of Fig. 4.

Also in this case we can observe an excellent match between simulation results and analytical predictions. Loss probability values are now higher than for the MI-NY channel because the link has smaller bandwidth, and the shorter RTT allows TCP sources to behave more aggressively.

D. Congestion on Both Bottlenecks

We now consider TCP connections for the transfer of Web pages from U.S. Web sites to users at Politecnico di Torino in the case when both the transoceanic link and the access link are congested. Notice that the link speeds are fairly different and, since 25% of the connections traverses both bottlenecks, and these are the considered connections, these results analyze a case of highly correlated bottlenecks.

In Fig. 8 we show results for the packet loss probability of those connections; the top plot presents analytical results and simulation point estimates versus N_1 and N_2 (the numbers of connections on the POLI-TO and MI-NY channels, respectively); the lower plot presents a two-dimensional (2-D) section for better readability. The plot is versus N_2 for different values of N_1; the other 2-D section is similarly accurate.

E. Results for Very Fat Pipes

One of the most attractive features of a useful analytical model is providing results for scenarios where simulation fails due to the excessive CPU or memory requirements. We found

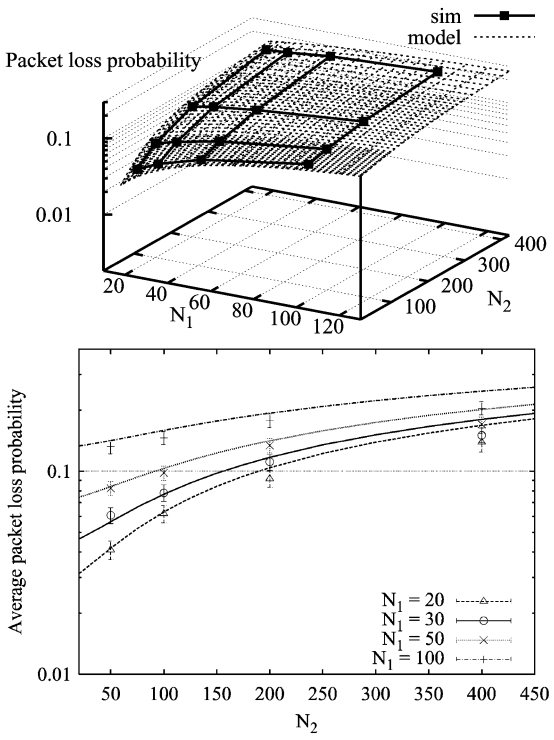

Fig. 8. Packet loss probability when connections cross both bottlenecks versus the number of connections N_1 and N_2 (upper plot), versus N_2 with constant N_1 (lower plot).

that running simulations with more than roughly 500 connections is very hard; however, channel speeds are growing fast and the number of concurrent connections follows closely. We explored with the model a not-too-futuristic scenario where the POLI-TO link capacity is 100 Mb/s and the MI-NY link capacity is 1 Gb/s (other links are scaled accordingly); simulation results are not available because the simulation model is too large to run on standard machines.

Figs. 9 and 10 report the three-dimensional (3-D) plots of the packet loss probability and the average window size in this scenario of fat pipes. The number of concurrent connections is up to 1000 on the POLI-TO link, and up to 12 000 on the MI-NY link, that collects most of the traffic between the U.S. and Northern Italy. The qualitative performance of the network does not change significantly by increasing the capacity and the number of connections. In particular, for a given bandwidth per connection value, the loss ratio and the average window size are almost constant. This observation strengthens the empirical model in [7] that derives the TCP throughput as a function of the loss probability. In addition, it partially relieves the worries concerning the possibility of exploiting very fast WAN networks with TCP connections, often expressed by researchers, but never proved due to the impossibility of simulating such environments or to load test beds with present-day applications.

F. Results for Individual Connections

The ergodic assumption inherent in any modeling technique, implies that the ensemble average over all connections coincides with the time average for a single connection with average characteristics. This property, together with the assumption that the average loss probability is equal for all connections sharing the

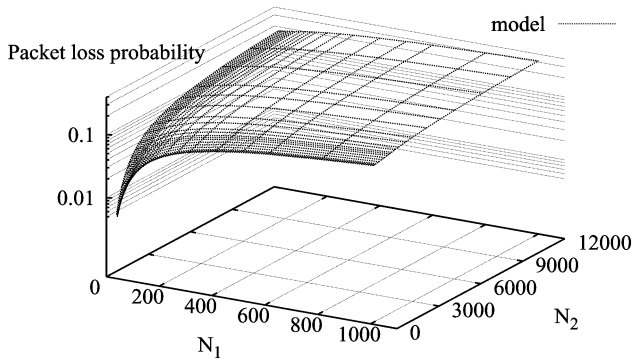

Fig. 9. Packet loss probability when connections cross both bottlenecks versus the number of connections N_1 and N_2; scenario with very fat pipes.

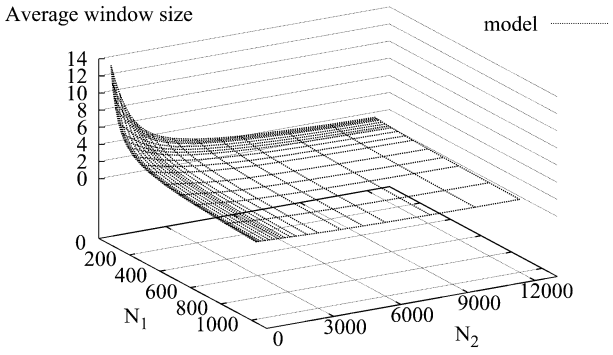

Fig. 10. Average window size when connections cross both bottlenecks versus the number of connections N_1 and N_2; scenario with very fat pipes.

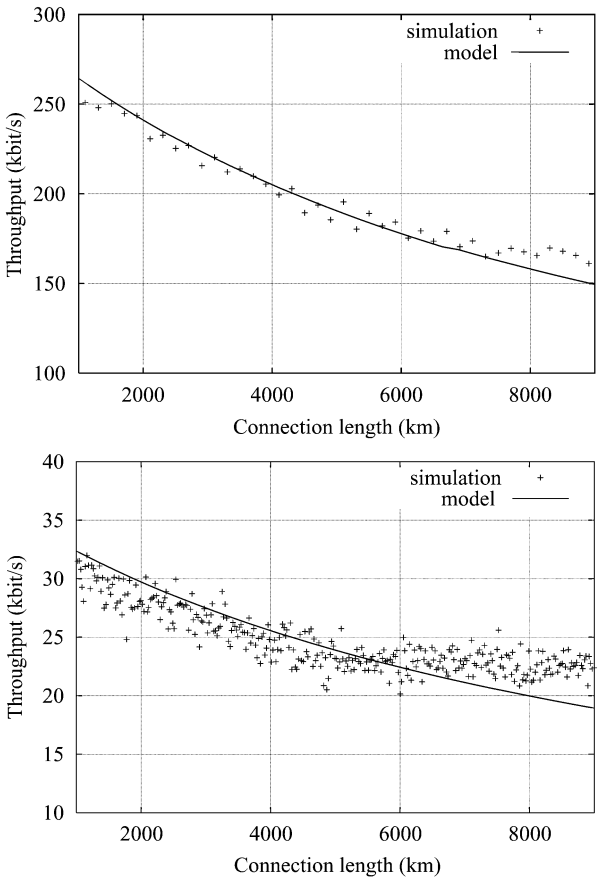

Fig. 11. Throughput of the individual connections as a function of their length for connections bottlenecked on the POLI-TO channel; case of 50 connections (upper plot) and 400 connections (lower plot).

same bottleneck link, allows the derivation of the performance of individual connections with the following procedure.

1) Solve the complete model with N concurrent connections and a given propagation delay distribution, deriving the average loss probability P_L and the average buffer occupancy $\overline{B}$.
2) Solve the TCP-submodel for one connection with its own propagation delay, using as fixed inputs P_L and $\overline{B}$ as computed in step 1. This solution yields the throughput, the window size distribution, the timeout probability and any other measure of interest for the individual connection.

The main limitation of the method lies in the connection length distribution used in step 1. Since the model solution is based on the use of the average propagation delay only, the best results are obtained when the coefficient of variation of connection lengths is not too large and no single connection is very different from the others.

Fig. 11 reports the throughput of TCP connections as a function of their length, obtained with the described procedure (solid line) and with simulation (crosses) assuming a uniform length distribution. The upper plot refer to a case with 50 competing connections, while the lower one to a case with 400 competing connections. The maximum precision of the model is reached for connections whose length is around the mean of the distribution, while for very short connections the throughput is slightly over-estimated and for very long ones it is slightly under-estimated. The relative error, however, never exceeds 15%. The error is due to the assumption of an equal average loss probability for all connections, that is not verified. Simulations show a linear decrease of the loss rate with the connections length. If the exact loss ratio is fed into the model, then the individual connection throughput is estimated very accurately, independently from the connection length. The same behavior is found for any number of competing connections.

G. A Meshed Topology

Up to now we have discussed a simple topology, where only two bottlenecks may arise. As explained in Section II-B2, the model can be used on any more complex and specific topology. In particular, we are interested here in assessing the model accuracy for a meshed topology where TCP groups merge and separate. Since the number of possible experimental setups explodes in a complex topology, we present results for a single meshed topology. Experiments on different networks yielded similar results.

Fig. 12 reports a generic meshed topology with seven nodes and five TCP groups transferring packets from source nodes S_i to receiver nodes R_i. All inter-router links have a propagation delay of 5 ms and the capacities as reported in the figure. Access links, i.e., links going from sources and receivers to the routers, have a propagation delay uniformly distributed between 1 and 10 ms. TCP groups follow the standard Internet min-hop routing, with group 2 breaking the tie in favor of the 80-Mb/s link.

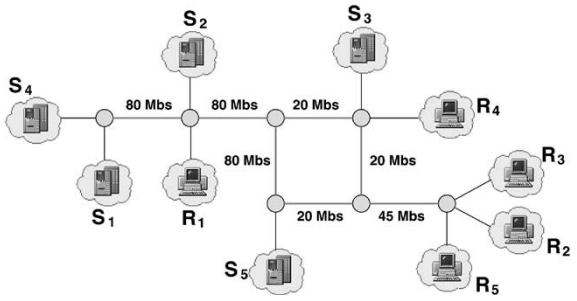

Fig. 12. Sample meshed topology with seven nodes and five TCP groups.

TABLE III
NUMBER OF CONNECTIONS PER TCP GROUP IN THE SIX DIFFERENT SETUPS EXPLORED ON THE TOPOLOGY IN FIG. 12

setup	group 1	group 2	group 3	group 4	group 5
1	100	25	25	50	50
2	50	10	50	100	25
3	30	10	20	40	10
4	200	20	10	25	100
5	10	50	100	200	30
6	400	100	200	10	20

Given the topology and the traffic relations (i.e., the TCP groups), the number of TCP connections within each group still leaves an enormous space to explore. Table III reports the six setups we consider here; each group has a variability over at least an order of magnitude through the six setups. In each setup, groups may compete for resources on one, two, or three links, depending on the load deriving from the specific setup.

Six setups times five TCP groups means that we have thirty different scenarios to analyze, that cannot be presented in detail. Thus, we show in Fig. 13 a compact representation of the validation results. The x-axis reports the results obtained with the model, while the y-axis reports the results obtained through simulation, with the relative confidence interval. If model and simulation match, the point lies on the bisector of the axes, shown by the straight line. The upper plot refers to the packet loss rate, while the lower one refer the the average connection throughput. The match of results is extremely satisfactory, specially for the throughput, where almost all points lie on the bisector.

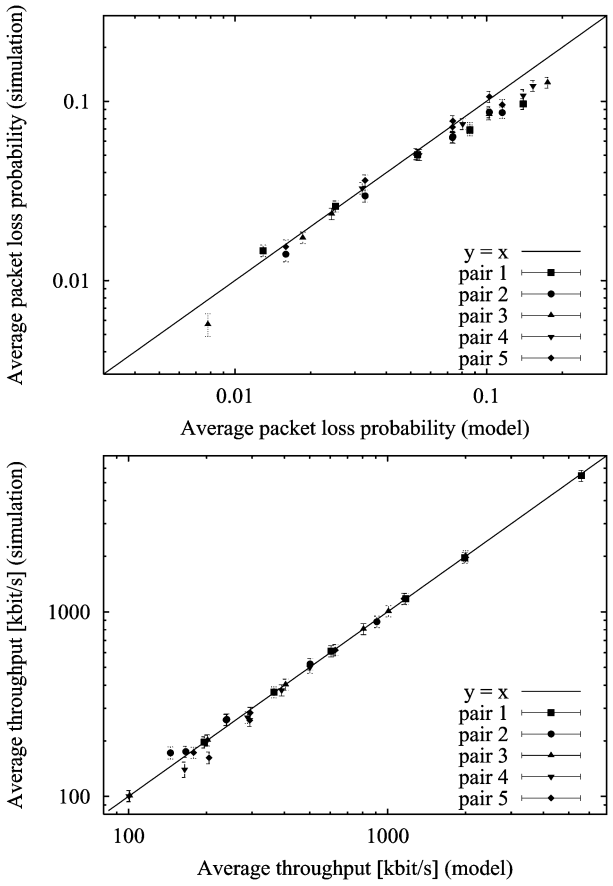

Fig. 13. Comparison between simulation and model results for the topology in Fig. 12 for the six setup in Table III.

IV. DISCUSSION AND CONCLUSION

In this paper, we have presented a closed queueing network model for the estimation of the performance of long-lived TCP connections. The queueing network provides a detailed description of the behavior of TCP-Tahoe connections and of their interaction with the underlying IP network. Numerical results have shown that the performance estimates provided by the closed queueing network model are extremely close to the performance predictions obtained from simulation experiments run with the *ns-2* package, in spite of the very limited computational cost for the solution of the analytical model.

The low complexity of the solution of the closed queueing network model yields a number of advantages; in particular, it allows:

- the study of more realistic networking setups with respect to the traditional single bottleneck network;
- the investigation of networks with thousands interacting TCP connections;
- the description of protocol details that cannot be normally incorporated into analytical models.

The model proved to be extremely robust in predicting the connections behavior under different network conditions, allowing the modeling of multiple bottlenecks with thousands of competing connections. The topology of the network can represent any chosen portion of the Internet or any particular "viewpoint" that users can have of the Internet itself. On the one hand, this second approach requires a preliminary analysis of the network to be modeled, in order to exclude from the network those parts than can never be congested. On the other hand, it allows the analysis of scenarios that represent meaningful setups as perceived, for instance, by users from an institution, that has a bottlenecked Internet access and whose traffic is directed mainly through a limited number of additional bottlenecks.

With a simple two-step procedure, it is also possible to extrapolate the behavior of individual connections as a function of the path they actually follow in the network. The only assumption that is necessary is that the average loss rate experienced at each single node by connections belonging to the same *group* is equal.

REFERENCES

[1] P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocols," *Comput. Commun. Rev.*, vol. 17, no. 5, pp. 2–7, Aug. 1987.

[2] R. Lo Cigno and M. Gerla, "Modeling window based congestion control protocols with many flows," *Perform. Eval.*, no. 36–37, pp. 289–306, Aug. 1999.

[3] M. Garetto, R. Lo Cigno, M. Meo, and M. A. Marsan, "A detailed and accurate closed queueing network model of many interacting TCP flows," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 1706–1715.

[4] A. Kumar, "Comparative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Trans. Networking*, vol. 6, pp. 485–498, Aug. 1998.

[5] A. Karnik and A. Kumar, "Performance of TCP congestion control with explicit rate feedback: Rate adaptive TCP (RATCP)," in *Proc. IEEE Globecom*, vol. 1, San Francisco, CA, Nov.–Dec. 2000, pp. 571–576.

[6] D. P. Heyman, T. V. Lakshman, and A. L. Neidhardt, "A new method for analysing feedback-based protocols with applications to engineering Web traffic over the Internet," in *Proc. ACM SIGMETRICS*, Seattle, WA, June 1997, pp. 24–38.

[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Trans. Networking*, vol. 8, pp. 133–145, Apr. 2000.

[8] T. Bonald, "Comparison of TCP Reno and TCP Vegas: Efficiency and fairness," *Perform. Eval.*, no. 36–37, pp. 307–332, Aug. 1999.

[9] A. Misra and T. Ott, "The window distribution of idealized TCP congestion avoidance with variable packet loss," in *Proc. IEEE INFOCOM*, New York, NY, Mar. 1999, pp. 1564–1572.

[10] A. Misra, T. Ott, and J. Baras, "The window distribution of multiple TCP's with random loss Queues," in *Proc. IEEE Globecom*, Rio de Janeiro, Brazil, Dec. 1999, pp. 1714–1726.

[11] A. Misra, J. Baras, and T. Ott, "Generalized TCP congestion avoidance and its effect on bandwidth sharing and variability," in *Proc. IEEE Globecom*, San Francisco, CA, Nov.–Dec. 2000.

[12] A. Abouzeid and S. Roy, "Analytic understanding of RED gateways with multiple competing TCP flows," in *Proc. IEEE Globecom*, vol. 1, San Francisco, CA, Nov.–Dec. 2000.

[13] V. Mishra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with and application to RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug.–Sept. 2000, pp. 151–160.

[14] C. V. Hollot, V. Mishra, W. B. Gong, and D. Towsley, "A control theoretic analysis of RED," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[15] E. Altman, F. Boccara, J. C. Bolot, F. Nain, P. Brown, D. Collange, and C. Fenzy, "Analysis of the TCP/IP flow control mechanism in high-speed wide-area networks," in *Proc. 34th IEEE CDC*, New Orleans, LU, Dec. 1995, pp. 368–373.

[16] F. Baccelli and D. Hong, "TCP is max-plus linear and what it tells us on its throughput," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug.–Sept. 2000, pp. 219–230.

[17] C. Casetti and M. Meo, "A new approach to model the stationary behavior of TCP connections," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 367–375.

[18] S. B. Fredj, T. Bonalds, A. Prutiere, G. Gegnie, and J. Roberts, "Statistical bandwidth sharing: A study of congestion at flow level," in *Proc. ACM SIGCOMM*, 2001, pp. 111–122.

[19] W. R. Stevens, *TCP/IP Illustrated*. Reading, MA: Addison-Wesley, 1994, vol. 1.

[20] Ns-2, Network Simulator (ver. 2). [Online]. Available: http://www.isi.edu/nsnam/ns/

[21] B. Braden, "Requirements for Internet hosts—Communication layers," IETF, RFC 1122, Oct. 1989.

[22] V. Jacobson, "Congestion avoidance and control," *Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, Aug. 1988.

[23] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.

Michele Garetto (S'00) received the Dr.Ing. degree in telecommunications engineering from the Politecnico di Torino, Torino, Italy, in May 2000, where he is currently working toward the Ph.D. degree in telecommunication networks.

From October 2001 to June 2002 he was with the Computer Science Department, University of Massachusetts, Amherst, as a Visiting Scholar. His research interests are in the field of performance evaluation of communication networks.

Renato Lo Cigno (A'92–M'03) received the Dr.Ing. degree in electronic engineering from the Politecnico di Torino, Italy, in 1988.

He is Associate Professor in the Department of Computer Science and Telecommunications (DIT), University of Trento, Italy, where he is one of the founding members of the Computer Networks research group. From 1999 to 2002, he was with the Electronics Department, Politecnico di Torino. From June 1998 to February 1999, he was with the Computer Science Department, University of California, Los Angeles, as a Visiting Scholar, working under grant CNR 203.15.8 from the Consiglio Nazionale delle Ricerche (CNR), the Italian National Research Council. His current research interests are in performance evaluation of wired and wireless networks, modeling and simulation techniques, and flow and congestion control.

Michela Meo (M'03) received the Dr.Ing. degree in electronic engineering and the Ph.D. degree in electronic and telecommunication engineering from the Politecnico di Torino, Italy, in 1993 and 1997, respectively.

Since 1997, she has been with the Telecommunication Networks Research Group, Politecnico di Torino, where she is currently an Assistant Professor. Her research interests are in the field of performance evaluation of communication networks with a particular focus on wireless systems and end-to-end performance.

Marco Ajmone Marsan (F'99) holds degrees in electronic engineering from the Politecnico di Torino, Torino, Italy, and the University of California at Los Angeles (UCLA). In 2002, he was awarded an *Honoris Causa* degree in telecommunication networks from the Budapest University of Technology and Economics, Budapest, Hungary.

He is a Full Professor in the Electronics Department, Politecnico di Torino, Italy, and the Director of the Institute for Electronics, Information Engineering and Telecommunications of the National Research Council. He has coauthored over 300 journal and conference papers in the areas of communications and computer science, as well as the two books, *Performance Models of Multiprocessor Systems* (Cambridge, MA: MIT Press) and *Modeling with Generalized Stochastic Petri Nets* (New York: Wiley). His current research interests are in the fields of performance evaluation of communication networks and their protocols.

Dr. Ajmone Marsan received the Best Paper Award at the Third International Conference on Distributed Computing Systems in Miami, FL, in 1982. He is a corresponding member of the Academy of Sciences of Torino. He participates in a number of editorial boards of international journals, including the IEEE/ACM TRANSACTIONS ON NETWORKING.